**IN THE CLAIMS**

Please amend the claims as follows:

1. (previously presented) A method of detecting one or more zombie global breakpoints for debugging computer software, said method including the steps of:

checking a breakpoint data structure to determine if the data structure has an entry for a breakpoint known to a debugging process for a certain address where a breakpoint fired;

if no entry is found by the checking of the data structure for the entry for the known breakpoint, verifying if a breakpoint condition continues to exist at the address where the breakpoint fired; and

if said breakpoint condition does not exist, identifying said breakpoint as a zombie breakpoint.

2. (original) The method according to claim 1, wherein said verifying step includes the step of checking that a special breakpoint instruction exists at said address, being the exception location.

3. (original) The method according to claim 1, wherein said verifying step includes the step of checking that an illegal breakpoint instruction exists at said address, being the exception location.

4. (original) The method according to claim 1, wherein said verifying step includes the step of checking that said address, being the exception location, is present in a special debug register.

5. (original) The method according to claim 1, wherein physical settings for causing a breakpoint exception at a particular location are detectable from a breakpoint handler.

6. (original) The method according to claim 5, wherein breakpoint removal logic is provided that lifts a physical breakpoint instruction from a breakpoint location before removing a breakpoint entry from said breakpoint data structure of said debugging process.

7. (previously presented) A computer-implemented apparatus for detecting one or more zombie global breakpoints for debugging computer software, said apparatus including:

a central processing unit for executing computer software;

memory for storing at least a portion of said computer software;

means for checking a breakpoint data structure to determine if the data structure has an entry for a breakpoint known to a debugging process for a certain address where a breakpoint fired;

means for, if no entry is found by the checking of the data structure for the entry for the known breakpoint, verifying if a breakpoint condition continues to exist at the address where the breakpoint fired; and

means for, if said breakpoint condition does not exist, identifying said breakpoint as a zombie breakpoint.

8. (original) The apparatus according to claim 7, wherein said verifying means includes means for checking that a special breakpoint instruction exists at said address, being the exception location.

9. (original) The apparatus according to claim 7, wherein said verifying means includes means for checking that an illegal breakpoint instruction exists at said address, being the exception location.

10. (original) The apparatus according to claim 7, wherein said verifying means includes means for checking that said address, being the exception location, is present in a special debug register.

11. (original) The apparatus according to claim 7, wherein physical settings for causing a breakpoint exception at a particular location are detectable from a breakpoint handler.

12. (original) The apparatus according to claim 11, wherein breakpoint removal logic is provided that lifts a physical breakpoint instruction from a breakpoint location before removing a breakpoint entry from said breakpoint data structure of said debugging process.

13. (previously presented) A computer program product having a computer readable medium having a computer program recorded therein for detecting one or more zombie global breakpoints for debugging computer software, said computer program product including:

computer program code means for checking a breakpoint data structure to determine if the data structure has an entry for a breakpoint known to a debugging process for a certain address where a breakpoint fired;

computer program code means for, if no entry is found by the checking of the data structure for the entry for the known breakpoint, verifying if a breakpoint condition continues to exist at the address where the breakpoint fired; and

computer program code means for, if said breakpoint condition does not exist, identifying said breakpoint as a zombie breakpoint.

14. (original) The computer program product according to claim 13, wherein said computer program code means for verifying includes computer program code means for checking that a special breakpoint instruction exists at said address, being the exception location.

15. (original) The computer program product according to claim 13, wherein said computer program code means for verifying includes computer program code means for checking that an illegal breakpoint instruction exists at said address, being the exception location.

16. (original) The computer program product according to claim 13, wherein said computer program code means for verifying includes computer program code means for checking that said address, being the exception location, is present in a special debug register.

17. (original) The computer program product according to claim 13, wherein physical settings for causing a breakpoint exception at a particular location are detectable from a breakpoint handler.

18. (original) The computer program product according to claim 17, wherein breakpoint removal logic is provided that lifts a physical breakpoint instruction from a breakpoint location before removing a breakpoint entry from said breakpoint data structure of said debugging process.